Robust visual tracking based on online learning sparse representation

Robust Visual Tracking Based on Online Learning Sparse Representation

Shengping Zhang[a], Hongxun Yao[a,*], Huiyu Zhou[b], Xin Sun[a], Shaohui Liu[a]

[a]*School of Computer Science and Technology, Harbin Institute of Technology, China*
[b]*Institute of Electronics, Communications and Information Technology, Queen's University Belfast, United Kingdom*

Abstract

Handling appearance variations is a very challenging problem for visual tracking. Existing methods usually solve this problem by relying on an effective appearance model with two features: 1) being capable of discriminating the tracked target from its background 2) being robust to the target's appearance variations during tracking. Instead of integrating the two requirements into the appearance model, in this paper, we propose a tracking method that deals with these problems separately based on sparse representation in a particle filter framework. Each target candidate defined by a particle is linearly represented by the target and background templates with an additive representation error. Discriminating the target from its background is achieved by activating the target templates or the background templates in the linear system in a competitive manner. The target's appearance variations are directly modeled as the representation error. An online algorithm is used to learn the basis functions that sparsely span the representation error. The linear system is solved via ℓ_1 minimization. The candidate with the smallest reconstruction error using the target templates is selected as the tracking result. We test the proposed approach using four sequences with heavy occlusions, large pose variations, drastic illumination changes and low foreground-background contrast. The proposed approach shows excellent performance in comparison with two latest state-of-the-art trackers.

Keywords: Visual tracking, sparse representation, online learning

1. Introduction

The purpose of visual tracking is to estimate the state of the tracked target in a video. It has wide applications such as intelligent video surveillance, advanced human computer interaction, robot navigation and so on. It is usually formulated as a search task where an appearance model is firstly used to represent the target and then a search strategy is utilized to infer the state of the target in current frame. Therefore, how to effectively model the appearance of the target and how to accurately infer the state from all candidates are two key steps for a successful tracking system. Although a variety of tracking algorithms have been proposed in the last decades, visual tracking still cannot meet the requirements of practical applications. The main difficulty of visual tracking is designing a powerful appearance model which should not only discriminate the target from its surrounding background but also be robust to its appearance variations. For the former issue, some promising progresses have been achieved recently by considering visual tracking as a two-class classification or detection problem. Many elegant features in the field of pattern recognition can be used to discriminate the target from its background. However, the latter is very difficult to achieve since there are a large number of un-predictive appearance variations over time such as pose changes, shape deformation, illumination changes, partial occlusion and so on. As shown in Fig. 1(a), the candidate marked by the blue rectangle is a "bad" candidate for tracking because there are a large number of background pixels inside it. An effective appearance model should not consider this candidate as the tracking result. In contrast, although the candidate marked by the red rectangle is partially occluded by the book, it is still a "good" candidate and should be considered as the tracking result. Existing appearance models in visual tracking cannot meet these requirements. For example, HSV his-

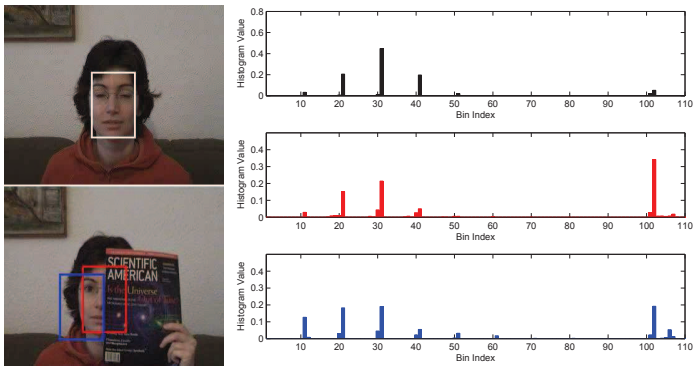

Figure 1: Illustrations of template and candidates. (a) The template is marked by white rectangle on the top. Two candidates are shown in the bottom, in which the"good" candidate and "bad" candidates are marked by red and blue rectangles, respectively. (b) The HSV histograms of the template, "good" candidate and "bad" candidate, respectively.

*Corresponding author
Email address: h.yao@hit.edu.cn (Hongxun Yao)

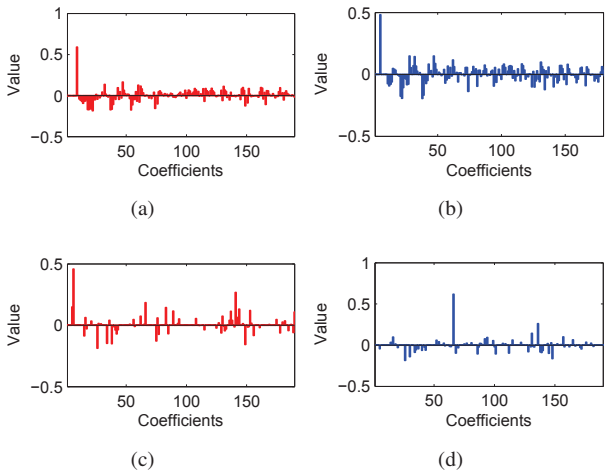

Figure 2: Coefficient examples. (a) The coefficient vector of the "good" candidate using identity matrix as error basis. (b) The coefficient vector of the "bad" candidate using identity matrix as error basis. (c) The coefficient vector of the "good" candidate using online learning error basis. (d) The coefficient vector of the "bad" candidate using online learning error basis.

togram was widely used to model the target's appearance [1]. We show the HSV histograms of the target template and two candidates in Fig. 1(b). Although modeling appearance with a HSV histogram can reflect the difference between the target template and the "bad" candidate, it also models the difference between the target template and the "good" candidate as shown in the tails of the three histograms. Furthermore, when certain similarity measure between two histograms is used as the data likelihood, the tracker may wrongly find the "bad" candidate as the final tracking result. For example, when Battacharyya coefficients are adopted to measure the similarity between two histograms, the similarity between the target template and the "bad" candidate is 0.852, which is larger than 0.829—the similarity between the target template and the "good" candidate. Therefore, traditional tracking methods that resort to an effective appearance model to achieve robust tracking are not always feasible. Recently, sparse representation has attracted much attention in the field of computer vision [2–4]. In [2], a robust face recognition method was proposed and the robustness to occlusions was achieved by introducing an error vector in the sparse representation model. Motivated by this idea, Mei and Ling proposed a robust visual tracking method based on sparse representation [3]. In their method, partial occlusion, appearance variances and other challenging issues were considered as the error vector represented by a set of trivial templates. We realized that both [2] and [3] used column vectors of the identity matrix as basis functions to linearly represent the error vector. These methods have some drawbacks. Firstly, the sparsity cannot be met. Each basis function used in their method can model whether one pixel position is occluded or not. It assumes that the occluded pixels occupy a relatively small portion of the entire image and the error vector has sparse nonzero entries. However, this assumption may not hold in real world especially when the target is severely occluded during tracking. For example, as shown in Fig. 1(a), the candidate marked by the red rectangle has almost half of the face occluded by the book. In this case, the representation coefficients are not sparse as shown in Fig. 2(a). Secondly, there are not basis vectors corresponding to background region. For example, the candidate marked by the blue rectangle in Fig. 1(a) contains some background pixels, its representation coefficients will also not be sparse as shown in Fig. 2(b). Finally, Mei's method uses a fixed basis to represent the error vector during the entire tracking. However, in practice, the error vector may change with the environment over time. Therefore, the fixed basis determined before tracking begins cannot effectively adapt the error vector to the environment's changes.

In this paper, motivated by the online basis learning for sparse coding [5], we proposed a novel tracking framework based on online learning sparse representation, which overcomes the aforementioned problems and achieves more robust results. The key idea of the proposed method is to simultaneously achieve discriminating the target from its background and being robust to its appearance variations separately based on sparse representation in a particle filter framework. Specifically, we sparsely represent each target candidate defined by a particle using target templates, background templates and error basis. The introduction of both the target and background templates in sparse representation is capable of discriminating the target from its background. For example, for the candidate corresponding to the target region, only target templates play key roles in the linear representation. In contrast, for the candidate corresponding to the background region, only background templates play roles in the linear representation. the target's appearance variants are modeled as the error in the linear representation. The error is spanned by a set of basis functions which are learned online when new observations are available over time. The representation coefficients are computed via ℓ_1 minimization. Each candidate is weighted in the particle filter framework based on the residuals when it is projected on the target templates. The tracking result is the weighted mean of all particles. The rationalities behind the proposed method are two-folds: Firstly, although appearance variations are unpredictable during tracking, they can still be compactly represented in certain subspace. Using the learned basis to represent the appearance variations can assure that the representation coefficients are sparse. Secondly, online basis learning can adaptively model unpredictable appearance variations during tracking, therefore improve the robustness of the tracking method.

The rest of the paper is organized as follows. In section 2 related methods on visual tracking and sparse representation are summarized. Section 3 details the tracking algorithm based on online learning sparse representation. Experimental results on four sequences are reported in section 4. We conclude this paper in section 5.

2. Related work

2.1. Visual tracking

In this subsection, in order to clarify the motivations of the proposed method, we reviewed the various appearance models

and inference methods for visual tracking.

2.1.1. Appearance modeling

Appearance modeling is used to represent the tracked target using information extracted from the target region. In the literature, widely used features are color [6], shape [7], texture [8], combination of color and texture [9], SIFT features [10], combination of color and SIFT [11] and attentional features [12]. For many tracking situations, color is a good choice for representing the tracked target because of its descriptive power and the fact that color information is readily accessible in image. Since color histogram was first proposed in [13], it has been successfully used to represent the target in visual tracking [1, 14] due to its simplicity, efficiency, and robustness. However, an evident limitation of color histogram is that it models the color distribution in a region but ignores the spatial relationship among pixels. Adding spatial information into color histogram improves the representation power. Along this line, color spatiogram [15] and color correlogram [16] are also employed for visual tracking. These methods model the target's appearance from a global perspective. When the target is locally occluded, the representation ability will significantly degrade. In order to overcome this shortcoming, in [17], a target candidate was divided into multiple patches and each one was represented by a color histogram. The robustness to occlusions was achieved by combining the vote maps of the multiple patches.

The appearance models mentioned above are fixed before tracking begins, which cannot effectively handle the target's appearance variations because such variations usually incur during tracking. Jepson and colleagues [18] proposed a framework to learn an adaptive appearance model, which adapts to the changing appearance over time. In [19], an online feature selection method was proposed to select features that are able to discriminate the target from its background. Since the feature selection is online when new observations are available, the selected features adapt to environment changes and achieve superior tracking results especially when the intensity difference between target and background is very small. In [20], a tracking method was proposed to incrementally learn a low-dimensional subspace representation, which efficiently adapts online to appearance changes. When the target suffered large changes in pose, scale and illumination, the method still accurately tracked the target. In [21], Kue *et. al.* proposed an AdaBoost based algorithm for learning a discriminative appearance model for multi-target tracking, which allows the models to adapt to target variations over time.

2.1.2. Inference methods

Given the appearance model of the target template, how to infer the target's state in current frame? Existing methods can be roughly classified into two classes: optimization and approximations. The first class performs a direct optimization procedure. The widely used optimization procedures are iterative gradient based search [22] and linear program [23]. The significant advantage of this method is its efficiency. However, the performance of optimization is not very stable especially in complex scenes. For example, once the tracker fails, it cannot recover from the lost. The second class uses probabilistic approximation to perform inference. Seminal work includes Kalman filter [24] and particle filter [1]. Particle filter has been successfully used in many tracking systems due to its efficiency and effectiveness.

In recent years, some novel tracking frameworks were proposed with impressive tracking performances. In [25], gait analysis was introduced to improve the tracking performance. In [26], visual tracking was embedded into a metric learning framework. Tracking-by-detection method [27] formulated visual tracking as a detection problem that detects whether the target appears or not in each candidate region. Many sophisticated machine learning technologies were introduced in this framework, e.g., semi-supervised classifier [28] and combining of supervised and semi-supervised classifiers [29].

2.2. Sparse representation

In this section, we review related work including applications of sparse representation in computer vision and the dictionary learning for sparse representation.

2.2.1. Applications of sparse representation

Recently sparse representation has been widely used in computer vision tasks, including face recognition [2], denoising and inpainting [30] and visual tracking [3, 4]. In these applications, the most related methods to us are [2] and [3]. In [2], a robust face recognition method was proposed, in which each test sample was linearly represented by all training samples. The recognition result of the test sample was encoded in the representation coefficients which can be solved by ℓ_1 minimization. The robustness to occlusion was achieved by introducing a representation error in the linear system. Compared to traditional face recognition methods, it obtains superior performance even with random features. Motivated by the success of sparse representation in face recognition, Mei and Ling proposed a visual tracking method based on ℓ_1 minimization [3]. Instead of representing each target candidate with target and trivial templates [3], in [4] a novel sparse representation framework was also proposed for visual tracking, which represents the target template with target candidates. The candidate with the largest coefficient was considered as tracking result.

2.2.2. Dictionary learning

Dictionary learning in sparse representation means to learn a dictionary from training data instead of using a predefined one. Since the learned dictionary can reflect the underlying structure of the data, it is more effective than the predefined one. The successful applications of learned dictionary in many visual systems [31, 32] have verified its excellent performance. Discriminative learning method was also used in dictionary learning for face recognition [33]. The learned dictionary cannot only sparsely represent each sample but also discriminate samples from different classes. In [5], an online dictionary learning method was proposed, which processes a small subset of the training set at a time. The computation cost is very low, which

is particularly important in the context of image and video processing [34].

3. Tracking based on online learning sparse representation

In this section, we give the details of the proposed tracking method based on online learning sparse representation. We briefly review the particle filter based tracking framework, and then formulate visual tracking as the sparse representation problem which simultaneously models the target and background templates as well as the error basis in a linear system, The online learning of the error basis and the template update are introduced, followed by a speed up strategy.

3.1. Particle filter tracking framework

The proposed algorithm is built upon on widely used tracking framework based on particle filter [1], which uses a set of weighted particles to approximate the posterior probability distribution of target's state. Due to its efficiency and robustness, excellent performance is obtained both in single object [1] and multiple target tracking [35]. Let $\boldsymbol{x}_t$ denote the target's state at time t. For example $\boldsymbol{x}_t$ can be a vector that consists of the coordinate and size of the tracked target [1]. Given a set of observed images $\boldsymbol{z}_{1:t} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_t\}$, particle filter can be used to infer the posterior probability $p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t})$ using a set of weighted samples $\{\boldsymbol{x}_t^i, \pi_t^i\}_{i=1:N}$ where $\boldsymbol{x}_t^i$ is the i-th particle at time t and π_t^i is the corresponding weights. The tracking result can be denoted as the weighted average of all particles $\frac{1}{N}\sum_{i=1}^{N}\pi_t^i\boldsymbol{x}_t^i$ or the particle with the largest weight.

3.2. Simultaneously modeling target, background and error in linear system

For each particle $\boldsymbol{x}_t^i$, we can obtain an image region using the coordinate and size in the particle. We then normalize the region to the predefined sizes $w \times h$. Gray values of all pixels in this region can be formed a 1D vector $\boldsymbol{y}_t^i \in \mathbb{R}^d$ ($d = w \times h$). In this paper, motivated by the success of sparse representation in computer vision [2, 3], we linearly represent each target candidate by a set of target and background templates. The target templates are normalized regions corresponding to the initially tracked target and updated during the tracking. The background templates are normalized regions around the initially tracked target and also updated over time. Specifically, at time t, given template set $\mathbf{P}_t = [\boldsymbol{p}_{t,1}, \ldots \boldsymbol{p}_{t,n}] \in \mathbb{R}^{d\times n}$, containing n_f target templates and $n - n_f$ background templates, a target candidate $\boldsymbol{y}_t^i$ can be approximately represented by $\mathbf{P}_t$ as

$$\boldsymbol{y}_t^i \approx \alpha_{t,1}^i\boldsymbol{p}_{t,1} + \alpha_{t,2}^i\boldsymbol{p}_{t,2} + \cdots + \alpha_{t,n}^i\boldsymbol{p}_{t,n} = \mathbf{P}_t\boldsymbol{\alpha}_t^i, \tag{1}$$

where $\boldsymbol{\alpha}_t^i = (\alpha_{t,1}^i, \alpha_{t,2}^i, \cdots, \alpha_{t,n}^i)^T \in \mathbb{R}^n$ is the coefficient vector of i-th candidate at time t. It should be noted that our linear system (Eq. 1) is different from [3] which only used target templates in the linear system. In our system, both target and background templates are used, which is able to handle more complex appearance variations. Intuitively, for a "good" target candidate, only target templates will be activated in the linear representation system. Similarly, for a "bad" target candidate, the coefficients corresponding to target templates tend to be zeros.

However, in practical scenarios, target's appearance will change drastically due to heavy occlusions, large pose variations, and drastic illumination changes. Previous methods focus on developing an effective appearance models that are robust to these variations. In this paper, we just use a simple appearance model (pixel intensity). The robustness to appearance variations is obtained by modeling appearance variations as the representation error as in [2, 3]

$$\boldsymbol{y}_t^i = \mathbf{P}_t\boldsymbol{\alpha}_t^i + \boldsymbol{\epsilon}_t^i, \tag{2}$$

where $\boldsymbol{\epsilon}_t^i$ is the representation error of the i-th candidate at time t. In [2, 3], they used identity matrix $\mathbf{I} \in \mathbb{R}^d$ as the basis set to span the error vector $\boldsymbol{\epsilon}_t^i$. In other words, they assume that the $\boldsymbol{\epsilon}_t^i$ can be represented in the pixel coordinates and this representation is sparse. We argue that this assumption is not valid when the appearance variations are significant. For example, when the target is occluded severely, the error $\boldsymbol{\epsilon}_t^i$ will not be sparse. The reason is that a larger proportion of pixels are occluded, accordingly, a larger proportion of basis vectors will be activated. Instead of using the identity matrix as the basis set, we adopt an online learned basis to span the error $\boldsymbol{\epsilon}_t^i$. Let $\mathbf{E}_t = [\boldsymbol{e}_{t,1}, \ldots, \boldsymbol{e}_{t,d}] \in \mathbb{R}^{d\times d}$ be the basis set learned at time t using online basis learning for sparse coding [5], then $\boldsymbol{\epsilon}_t^i$ can be sparsely represented by

$$\boldsymbol{\epsilon}_t^i \approx \beta_{t,1}^i\boldsymbol{e}_{t,1} + \beta_{t,2}^i\boldsymbol{e}_{t,2} + \cdots + \beta_{t,d}^i\boldsymbol{e}_{t,d} = \mathbf{E}_t\boldsymbol{\beta}_t^i, \tag{3}$$

where $\boldsymbol{\beta}_t^i = [\beta_{t,1}^i, \beta_{t,2}^i, \cdots, \beta_{t,d}^i]^T \in \mathbb{R}^d$ is the sparse coefficient vector. Then the target candidate can be represented as

$$\boldsymbol{y}_t^i = \mathbf{P}_t\boldsymbol{\alpha}_t^i + \boldsymbol{E}_t\boldsymbol{\beta}_t^i = \begin{bmatrix}\mathbf{P}_t & \mathbf{E}_t\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha}_t^i \\ \boldsymbol{\beta}_t^i\end{bmatrix} = \mathbf{B}_t\boldsymbol{c}_t^i, \tag{4}$$

where $\mathbf{B}_t = \begin{bmatrix}\boldsymbol{P}_t & \boldsymbol{E}_t\end{bmatrix} \in \mathbb{R}^{d\times(n+d)}$ is the basis set and $\boldsymbol{c}_t^i = \begin{bmatrix}\boldsymbol{\alpha}_t^i \\ \boldsymbol{\beta}_t^i\end{bmatrix} \in \mathbb{R}^{n+d}$ is the coefficient vector. It also should be noted that although the form of Eq. 4 is similar with the one in [2, 3], the intrinsic meanings are different because we use the online learned basis instead of the column vectors of the identity matrix. Since $n \ll d$ and $\boldsymbol{\beta}_t^i$ is sparse, the coefficient vector $\boldsymbol{c}_t^i$ is sparse and can be obtained by ℓ_1 minimization

$$\hat{\boldsymbol{c}}_t^i = \arg\min_{\boldsymbol{c}_t^i} \|\boldsymbol{c}_t^i\|_1 \quad \text{subject to} \quad \boldsymbol{y}_t^i = \mathbf{B}_t\boldsymbol{c}_t^i. \tag{5}$$

In Fig. 2, Fig. 2(c) and Fig. 2(d) are the obtained coefficient vectors with size of 190 of the "good" and "bad" candidates in our method. Their numbers of nonzero coefficients are 65 and 70, respectively, which shows that the sparsity of the coefficient vector obtained by our learned basis is guaranteed. In contrast, if the identity matrix is used as the basis set in [3], the obtained coefficient vectors are not sparse as shown in Fig. 2(a) and Fig. 2(b) where the number of nonzero coefficients of the two coefficient vectors are 156 and 160, respectively.

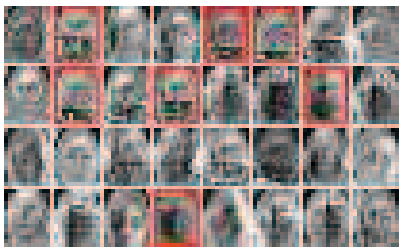

Figure 3: In this figure, we show 32 basis vectors out of total of 190 basis vectors learned on the *face* sequence by our method. Some basis vectors (marked by red rectangle) reflect how the face is occluded by the book. Note that all vectors are scaled for visualization.

Instead of projecting the candidate only on the target templets as in [3], we compute the residual of the candidate $\boldsymbol{y}_t^i$ after it was projected on the target templates as well as error basis

$$r(\boldsymbol{y}_t^i) = \|\boldsymbol{y}_t^i - \mathbf{P}_t\delta_f(\boldsymbol{\alpha}_t^i) - \mathbf{E}_t\boldsymbol{\beta}_t^i\|_2, \tag{6}$$

where $\delta_f(\boldsymbol{\alpha}_t^i)$ is a new vector whose only nonzero entries are the entries in $\boldsymbol{\alpha}_t^i$ that are associated with target templates.

The "best" target candidate is the one that has the smallest residual with index $\Im = \arg\min_i r(\boldsymbol{y}_t^i)$. The "worst" target candidate is the one that has the largest residual with index $\aleph = \arg\max_i r(\boldsymbol{y}_t^i)$. Then the tracking result at current time is the particle $\boldsymbol{x}_t^{\Im}$ which corresponds to the "best" target candidate.

3.3. Template update and online basis learning

In this paper, we adopt a simple weight based update strategy. We weight all target template according to their roles in representing the "best" target candidate. Similarly, all background templates are weighted according to their roles in representing the "worst" target candidate. Specifically, we weight j-th target template with its coefficient $\boldsymbol{\alpha}_{t,j}^{\Im}$, $j = 1, \ldots, n_f$ when representing the "best" target candidate. Then we sort all target templates in descending order according to their weights. We weight j-th background template with its coefficient $\boldsymbol{\alpha}_{t,j+n_f}^{\aleph}$, $j = 1, \ldots, n-n_f$ when represent the "worst" target candidate. All background templates are sorted in descending order according to their weights. The target template with the lowest weight is updated using the "best" candidate as

$$\boldsymbol{y}_t^{n_f} = \boldsymbol{y}_t^{n_f} + (1-\eta)\boldsymbol{y}_t^{\Im}, \tag{7}$$

where η is the update rate parameter. The background template with the lowest weight is updated using the "worst" candidate as

$$\boldsymbol{y}_t^{n} = \boldsymbol{y}_t^{n} + (1-\eta)\boldsymbol{y}_t^{\aleph}. \tag{8}$$

We model the target's appearance variations by introducing an error vector in Eq. 2. In order to assure that the coefficient vector $\boldsymbol{\beta}_t^i$ is sparse and adaptive to target's appearance variations over time, we use the online basis learning algorithm [5] to learn the error basis set $\mathbf{E}_t$. Specifically, at time t, the target template with the largest weight is $\boldsymbol{p}_t^1$. We collect training samples $\{\boldsymbol{s}_t^i = \boldsymbol{y}_t^i - \boldsymbol{p}_t^1\}_{i=1:N}$. The $i-th$ sample is represented by all column vectors of $\mathbf{E}_{t-1}$ and the representation coefficient vector $\boldsymbol{\alpha}_t^i$ can be computed by sparse coding model [36]. The current basis set $\mathbf{E}_t$ can be obtained by minimizing the cost function

$$C(\mathbf{E}_t) = \sum_{i=1}^{N} \|\boldsymbol{s}_t^i - \mathbf{E}_t\boldsymbol{a}_t^i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1 \tag{9}$$

The detailed steps of solving this minimization problem can be found in [5]. An illustration of some learned basis vectors on the *face* sequence are shown in Fig. 3, from which we can see that some basis vectors (e.g., marked by red rectangle) reflect how the face is occluded by the book.

3.4. Reducing computation time

In practical tracking application, the linear system Eq. 4 is very large. For instance, if the normalized size of the target is 64×64, the dimension of the feature vector $\boldsymbol{y}_t^i$ is on the order of 10^3. Although solving Eq. 5 relies on scalable methods such as linear programming, the computation cost is still beyond the capability of normal computers. In [37], a rapid face recognition based on sparse representation is proposed, where the computation time reduction is achieved by applying a hash matrix to both sides of the linear system. In this paper, we also utilize the hash matrix to reduce the computation cost of the proposed method.

Let $s \in \{1, \ldots, S\}$ be the seed, we can define a hash function $h_s(j, d) : \mathbb{N} \to \{1, \ldots, d\}$. The hash matrix $\mathbf{H} = (H_{ij})$ is then defined as in [37]

$$H_{ij} = \begin{cases} 2h_s(j,2) - 3, & h_s(j,d) = i, \forall s \in \{1,\ldots,S\} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Applying $\mathbf{H}$ to both sides of Eq. 4 yields

$$\mathbf{H}\boldsymbol{y}_t^i = \mathbf{H}\mathbf{B}_t\boldsymbol{c}_t^i \tag{11}$$

which can be solved the following ℓ_1 minimization

$$\hat{\boldsymbol{c}}_t^i = \arg\min_{\boldsymbol{c}_t^i} \|\boldsymbol{c}_t^i\|_1 \quad \text{subject to} \quad \mathbf{H}\boldsymbol{y}_t^i = \mathbf{H}\mathbf{B}_t\boldsymbol{c}_t^i \tag{12}$$

The detailed algorithm is summarized in Algorithm 1.

4. Experimental results

In order to validate the effectiveness of the proposed method, we conduct experiments on four challenging sequences which involve severe appearance variations including heavy occlusions, large pose variations, and drastic illumination changes as well as low foreground background contrast. The proposed method is compared with two latest state-of-the-art methods named Incremental Visual Tracking (IVT) [20] and ℓ_1 minimization tracking (L1) [3]. Note that for all test sequences, we set main parameters as: $\eta = 0.7$, $w_0 = 0.05$, $N = 200$, $n_f = 10$ and $n = 20$. Noted that all these parameters are set by hand tuning with some prior knowledge. Taking η as an example, this parameter controls how much the new date impacts the template. We think the new date can only contribute a small part to

Algorithm 1: Tracking based on online learning sparse representation

Input: $\boldsymbol{x}_0^* \in \mathbb{R}^6$ (initial state), $\boldsymbol{y}_0^* \in \mathbb{R}^d$ (initiate template), $\mathbf{E}_0 = \mathbf{I} \in \mathbb{R}^{d\times d}$ (initial basis set), Σ_* (Gaussian covariance), η (template update rate), w_0 (initial weight), T (length of sequence), N (number of particles), n (number of templates), n_f (number of target templates)

Initialize particles set: $\{\boldsymbol{x}_0^i = \boldsymbol{x}_0^*, \pi_0^i = \frac{1}{N}\}_{i=1:N}$ and template set $\{\boldsymbol{p}_{0,i} = \boldsymbol{y}_0^*\}_{i=1:n}$
for $t = 1$ **to** T **do**
  for $i = 1$ **to** N **do**
    Extract target candidate $\boldsymbol{y}_t^i$
    Compute sparse coefficients $\boldsymbol{\alpha}_t^i$ and $\boldsymbol{\beta}_t^i$ by solving Eq. 12 via ℓ_1 minimization
    Compute residual $r(\boldsymbol{y}_t^i) = \|\boldsymbol{y}_t^i - \mathbf{P}_t\delta_f(\boldsymbol{\alpha}_t^i) - \mathbf{E}_t\boldsymbol{\beta}_t^i\|_2$
    Weight i-th particle with $\pi_t^i = \exp(-\lambda r(\boldsymbol{y}_t^i))$
    Collect sample $\boldsymbol{y}_t^i - \boldsymbol{p}_t^1$
  end
  Update basis $\mathbf{E}_t$ with samples $\{\boldsymbol{s}_t^i = \boldsymbol{y}_t^i - \boldsymbol{p}_t^1\}_{i=1:N}$ using Eq. 9
  Compute index of the "best" candidate as $\Im = \arg\min_i r(\boldsymbol{y}_t^i)$
  Compute index of the "worst" candidate as $\aleph = \arg\max_i r(\boldsymbol{y}_t^i)$
  Weight j-th target template with $\hat{\alpha}_{t,j}^{\Im}$, $j = 1, \ldots, n_f$
  Weight j-th background template with $\hat{\alpha}_{t,j+n_f}^{\aleph}$, $j = 1, \ldots, n - n_f$
  Sort target and background templates according to their weights, respectively
  Update target template as $\boldsymbol{y}_t^{n_f} = \boldsymbol{y}_t^{n_f} + (1-\eta)\boldsymbol{y}_t^{\hat{i}}$, with initial weight w_0
  Update background template as $\boldsymbol{y}_t^n = \boldsymbol{y}_t^n + (1-\eta)\boldsymbol{y}_t^{\tilde{i}}$, with initial weight w_0
  return $\boldsymbol{x}_t^{\Im}$ as the tracking result
  $\mathbf{E}_{t+1} \leftarrow \mathbf{E}_t$
  Resample particle set according to their weights and transit particles to time $t+1$ by sampling from a Gaussian distribution $\boldsymbol{x}_{t+1}^i \sim \mathcal{N}(\boldsymbol{x}_t^i, \Sigma^*)$
end

the template, so we set the parameter to 0.7. The comparison is performed from both visual evaluation and quantitative evaluation in 4.1 and 4.2, respectively. The performance explanations are given in 4.3. We also test the running speed of the proposed method in 4.4.

4.1. Visual evaluation

The first sequence is the *doll* sequence with 430 frames obtained from [20]. An animal doll moves with significant pose, lighting and scale variation in a cluttered scene. The normalization size is 14×14. In order to analyze the details of the proposed method, we show some intermediate results of the 50th frame of this sequence. As shown in Fig. 8, some candidates are similar to the tracked targets. However, others have significant difference with the tracked target. A good tracking algorithm should distinguish these candidates and find the most similar candidate as the tracking result. In Fig. 9, we show all foreground and background templates. It should be noted that in this work, we mean the background templates as the image regions containing some background pixels rather than those background regions. The reason is that in a particle filter framework, although most particles locate at the target region, there are still some particles around the target region, which will causes candidates corresponding to these particles will contain background pixels.

Figure 8: All candidates corresponding to 200 particles at time 50 on *doll* sequence.

Some samples of the final tracking results are shown in Fig. 4. The frame indices are 50, 100, 200, 300, 396 and 430. From Fig. 4, we can see that our tracker is capable of tracking the doll all the time even when it changes pose drastically. The IVT tracker also achieves comparable performance. However, the L1 tracker fails to track the target in the fifth index frame and does not recover later.

The second test sequence is the *face* sequence [17] with 300 frames. The face of the women is severely occluded by a book. The normalization size is 15×12. Six representative frames with indices 20, 47, 89, 140, 180, 280 are shown in Fig. 5 where rows 1, 2 and 3 are for our proposed tracker, L1 tracker and IVT

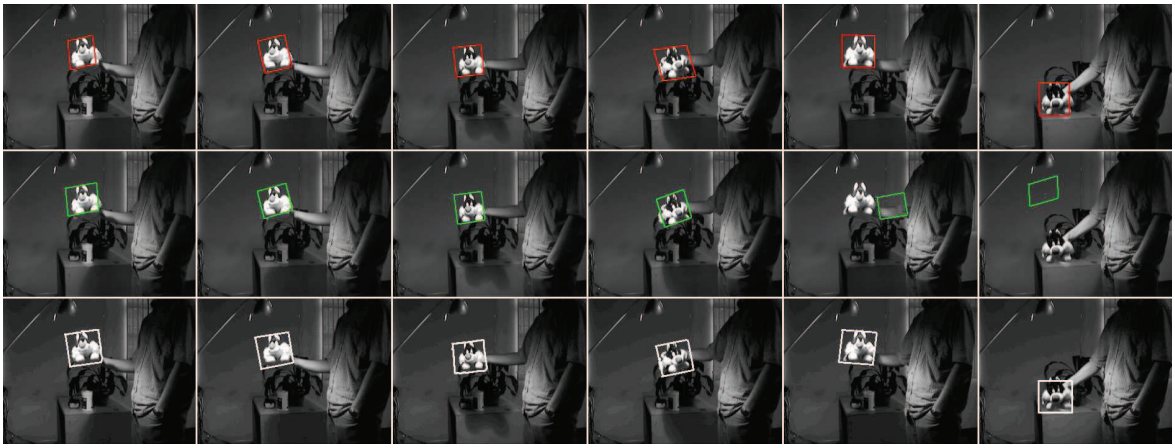

Figure 4: The tracking results of the *doll* sequence with significant pose, lighting and scale variation in a cluttered scene. The first, second and third rows are results obtained by the proposed method, L1 tracker and IVT tracker, respectively.

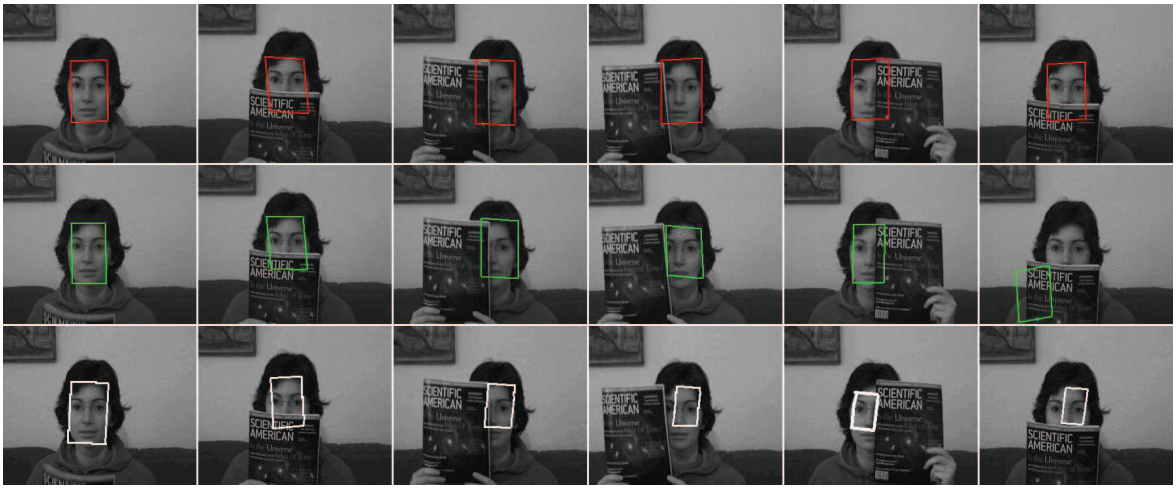

Figure 5: The tracking results of the *face* sequence with severe occlusions. The first, second and third rows are results obtained by the proposed method, L1 tracker and IVT tracker, respectively.

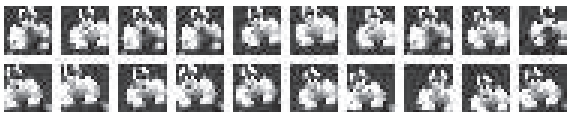

Figure 9: all foreground and background templates at time 50 on *doll* sequence. The top row shows the 10 foreground templates and the bottom row shows the 10 background templates.

tracker, respectively. Due to the presence of occlusions, IVT tracker drifts to the un-occluded face region. The L1 tracker obtains the similar performance compared with the proposed method before 150th frame. However, it loses the target soon after. In contrast, the proposed method successfully track the face in the entire sequence even with severe occlusions by the book.

In order to evaluate our tracker in outdoor environments, where lighting conditions often change drastically, we conduct experiment on the *head* sequence with 240 frames from [20]. In it, a person walks underneath a trellis covered by vines, resulting in significant appearance variations of his head due to cast shadows. Frames with indices 50, 100, 150, 183, 200 and 218 are shown in Fig. 6. It can be observed that our tracker is able to track the target accurately. However, the L1 and IVT trackers drift apart when the significant illumination happens in the 150th frame.

In addition to appearance variations, low foreground-background contrast is also very difficult for visual tracking. We test whether the proposed method can overcome this difficulty or not on the *PkTest02* sequence from VIVID benchmark dataset [38]. It is an infrared image sequence where the target-to-background contrast is very low. We give the tracking results on six representative frames with indices 20, 41, 60, 80, 100 and 120 in Fig. 7, which shows that our tracker is capable of tracking the car all the time even with severe occlusions by the

Figure 6: The tracking results of the *head* sequence with significant pose, lighting and scale variation in a cluttered scene. The first, second and third rows are results obtained by the proposed method, L1 tracker and IVT tracker, respectively.

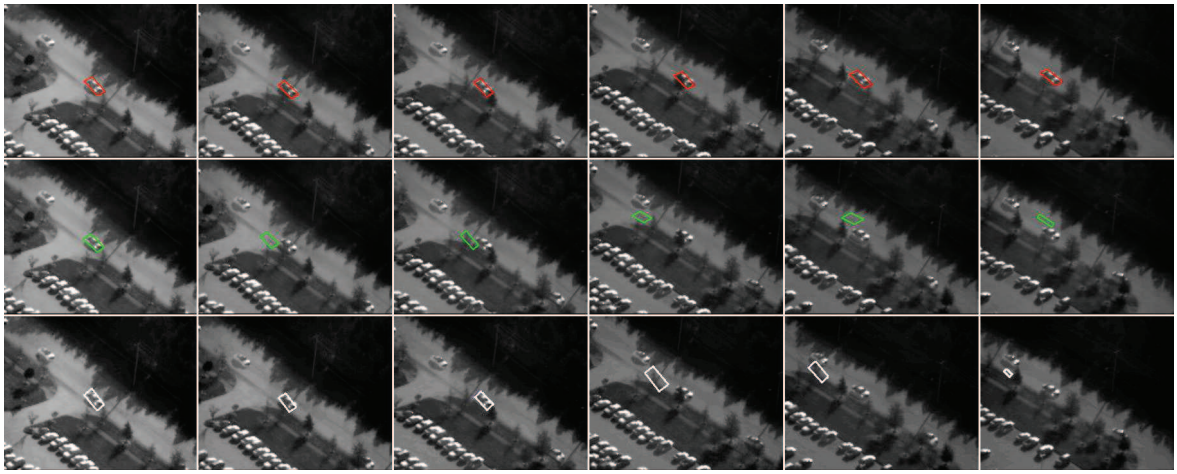

Figure 7: The tracking results of the *PkTest02* sequence with significant pose, lighting and scale variation in a cluttered scene. The first, second and third rows are results obtained by the proposed method, L1 tracker and IVT tracker, respectively.

trees on the roadside. In comparison, the L1 tracker loses the car when it is first occluded by a tree in the 20th frame. The IVT tracker is superior to the L1 tracker and successfully track the car in the 20th frame. However, it finally fails to track the car in the 80th frame due to more severe occlusions by a bigger tree.

4.2. Quantitative evaluation

In order to evaluate the overall performance of the proposed method, we also perform quantitative evaluation which is based on the tracking error e in each frame. We use a simple measure $e = \epsilon/d$, where ϵ is the offset of the center of the tracking result from the ground truth and d is the diagonal length of the tracking result. The smaller e is, the better the tracking performance is obtained. In Fig. 10, we present the tracking error-time curves of the four test sequences. We can see that the overall performance of our tracker is significant superior to the L1 and IVT trackers on all sequences, which validate the robustness of the proposed method.

4.3. Performance explanations

The experimental results above show that our tracking algorithm is superior to both L1 and IVT trackers. Here, we will provide further discussions. The IVT tracker incrementally learns a low-dimensional subspace used to represent the target. This representation is online updated as new observations arrive. The tracker can adapt to the target's appearance variations to some extent. However, when the target's appearance changes drastically, the subspace learned from the historic observations cannot effectively represent the current target. Comparably, the L1 tracker separates the appearance modeling from the variation representation. The appearance model consists of a set of target templates. Variations are represented by a set of trivial templates. Since the trivial templates are columns of the identity matrix, the representation coefficients can be obtained correctly via ℓ_1 minimization when their variations are minor.

However, when the variations become severe, the sparsity of coefficients cannot be guaranteed. So coefficients obtained by ℓ_1 minimization in this case cannot estimate whether a candidate is the target template or not, leading to the tracking failure.

Unlike the IVT and L1 trackers, our tracker overcomes the weaknesses aforementioned: First, instead of using a fixed basis like the L1 tracker, we learn the basis online which assures that this representation can more efficiently adapt to appearance changes than the L1 tracker. In this aspect, our method is also significantly different with IVT tracker although both of them online learn basis. The IVT tracker learns the basis used to represent the target's appearance. However, our tracker learns the basis used to represent the appearance's variations. Intuitively, our tracker can be more effective than IVT tracker when handling target's appearance variations. Second, both the IVT and L1 trackers cannot effectively discriminate the target from the background and do not consider the background information in their models. In contrast, we introduce appropriate background templates in our sparse representation. The background templates and the target templates coherently represent each candidate, whilst the representation coefficients are obtained by ℓ_1 minimization. Our tracker can discriminate the target from the background well due to the different distributions of the target and background templates when used to represent the candidate. In summary, our tracker has two features: 1)online learning of the target's appearance variations and 2)discrimination capability of detecting the target from the background, which ensure the outstanding performance of the proposed method.

4.4. Processing speed

The proposed method is implemented in Matlab on a Pentium-IV 3GHz PC with 1G RAM. We test the processing speed of our tracker on *face* sequence which consists of 300 frames of size 352×288. The average processing speed of the our tracker is about 5 frames per second, which will be further improved with code optimization.

5. Conclusion

In this paper we propose a robust visual tracking algorithm based on online learning sparse representation. The main contribution is that we integrate two requirements of an expected appearance model, discriminating the tracked target from background and being robust to appearance variations, into a linear representation system that each target candidate is represented by target templates, background templates and online learned error basis. To the best of our knowledge, it is the first time that these two requirement are achieved separately based on sparse representation. Experimental results of four challenging sequences validate that the proposed method is significantly superior to two latest state-of-the-art methods.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No.: 61071180 and Key Program Grant No.: 61133003). Shengping Zhang is also supported by funding of Ph.D. student short-term visiting abroad from HIT. Huiyu Zhou is currently supported by UK EPSRC Grant EP/G034303/1 and Invest NI.

References

[1] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," *Proceedings of European Conference on Computer Vision*, pp. 661–675, 2002.

[2] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[3] X. Mei and H. Ling, "Robust visual tracking using L1 minimization," *Proceedings of the 12th International Conference on Computer Vision*, pp. 1436–1443, 2009.

[4] S. Zhang, H. Yao, X. Sun, and S. Liu, "Robust object tracking based on sparse representation," *Proceedings of SPIE International Conference on Visual Communications and Image Processing*, pp. 77 441N–1–8, 2010.

[5] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," *Proceedings of the 26th Annual International Conference on Machine Learning*, vol. 382, pp. 689–696, 2009.

[6] G. Bradski, "Computer vision face tracking as a component of a perceptual user interface," *Intelligence Technology Journal*, vol. 2, pp. 1–15, 1998.

[7] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *Proceedings of the 4th European Conference on Computer Vision*, pp. 343–356, 1996.

[8] A. Shahrokni, T. Drummond, and P. Fua, "Fast texture-based tracking and delineation using texture entropy," *Proceedings of International Conference on Computer Vision*, vol. 2, pp. 1154–1160, 2005.

[9] S. Zhang, H. Yao, and S. Liu, "Partial occlusion robust object tracking using an effective appearance model," *Proceedings of SPIE International Conference on Visual Communications and Image Processing*, pp. 77 442U–1–8, 2010.

[10] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.

[11] S. Zhang, H. Yao, and P. Gao, "Robust object tracking combining color and scale invariant features," *Proceedings of SPIE International Conference on Visual Communications and Image Processing*, pp. 77 442R–1–8, 2010.

[12] S. Zhang, H. Yao, and S. Liu, "Robust visual tracking using feature-based visual attention," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1150–1153, 2010.

[13] M. Swain and D. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[14] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.

[15] S. Birchfield and R. Sriram, "Spatiograms versus histograms for region-based tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 1158–1163, 2005.

[16] Q. Zhao and H. Tao, "Object tracking using color correlogram," *Proceedings of IEEE Workshop Performance Evaluation of Tracking and Surveillance*, pp. 263–270, 2005.

[17] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 798–805, 2006.

[18] A. Jepson, D. Fleet, and T. EI-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1296–1311, 2003.

[19] R. T. Collins, Y. Liu, and M. Leordeanu, "On-line selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631–1643, 2005.

[20] D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 8, pp. 125–141, 2007.

[21] C. Kuo, C. Huang, and R. Nevatia, "Multi-target tracking by on-line

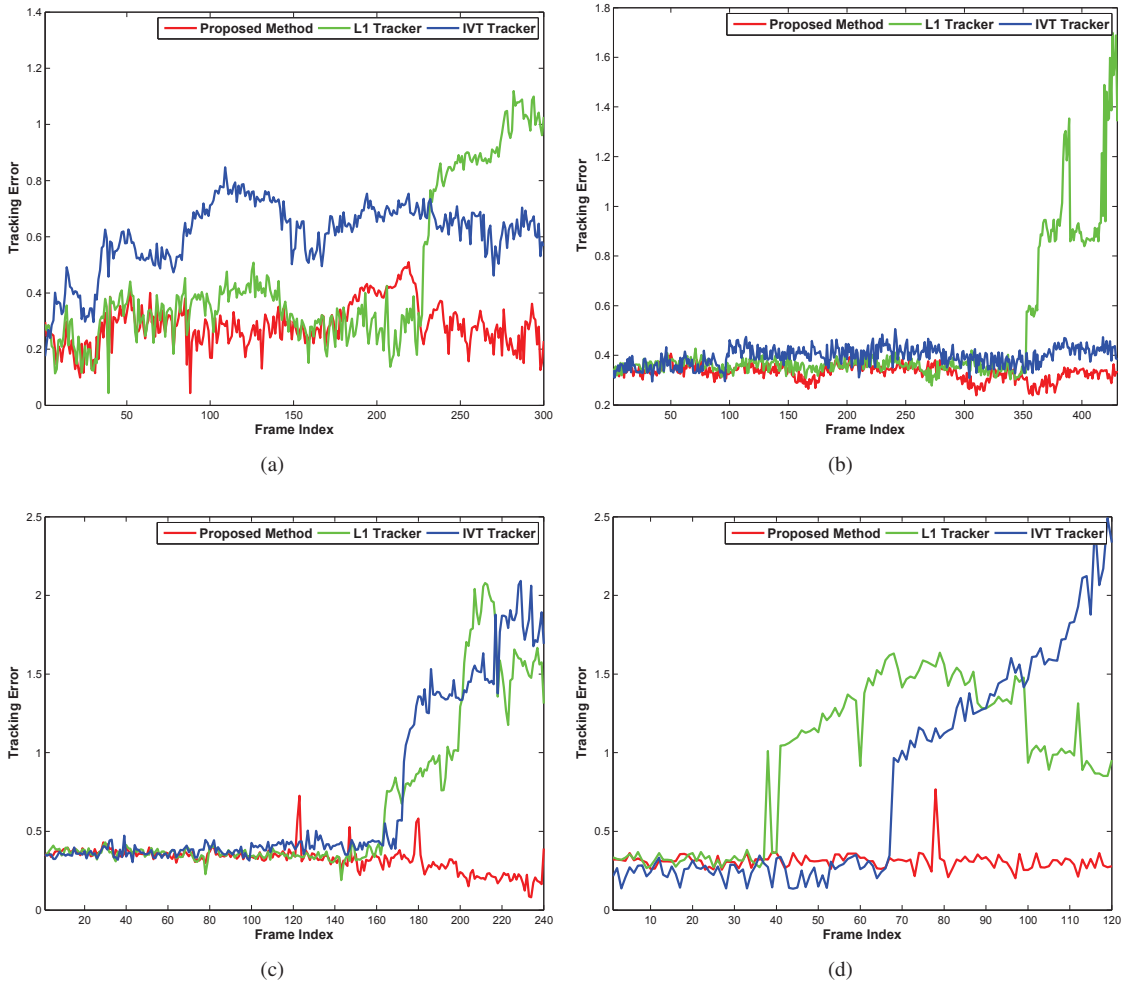

Figure 10: Quantitative evaluations on (a) *face* sequence, (b) *doll* sequence, (c) *head* sequence and (d) *PkTest02* sequence.

learned discriminative appearance models," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010.

[22] Q. Zhao, S. Brennan, and H. Tao, "Differential EMD tracking," *Proceedings of IEEE Conference on Computer Vision*, pp. 1–8, 2007.

[23] Y. Wu and J. Fan, "Contextual flow," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 33–40, 2009.

[24] P. Gutman and M. Velger, "Tracking targets using adaptive kalman filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 5, pp. 691–699, 1990.

[25] H. Zhou, A. Wallace, and P. Green, "Efficient tracking and ego-motion recovery using gait analysis," *Signal Processing*, vol. 89, no. 12, pp. 2367–2384, 2009.

[26] X. Wang, G. Hua, and T. Han, "Discriminative tracking by metric learning," *Proceedings of the 11th European Conference on Computer Vision*, pp. 200–214, 2010.

[27] H. Grabner and H. Bischof, "On-line boosting and vision," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 260–267, 2006.

[28] H. Grabner and C. Leistner, "Semi-supervised on-line boosting for robust tracking," *Proceedings of the 10th European Conference on Computer Vision*, pp. 234–247, 2008.

[29] S. Stalder, H. Grabner, and L. van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," *Proceedings of IEEE Conference on Computer Vision workshop*, pp. 1409–1416, 2009.

[30] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," *Proceedings of IEEE Conference on Computer Vision*, pp. 2272–2279, 2009.

[31] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[32] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," *Proceedings of Advances in Neural Information Processing Systems*, pp. 1033–1040, 2008.

[33] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010.

[34] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, 2009.

[35] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2009.

[36] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[37] Q. Shi, H. Li, and C. Shen, "Rapid face recognition using hashing," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2010.

[38] R. Collins, X. Zhou, and S. K. Teh, "An open source tracking testbed and evaluation web site," *IEEE Workshop Performance Evaluation of Tracking and Surveillance*, pp. 1–8, 2005.